

IN THE CLAIMS

Please amend the claims as indicated:

1. (currently amended) A method for language verification of a Java card CAP file created from an original Java code file, comprising:

a) a conversion step for converting said Java card CAP file into a corresponding converted Java code file that is semantically identical to said Java card CAP file, wherein the Java card CAP file is created from an the original file contains classes that are capable of being compiled, and wherein the only executable instructions in the Java card CAP file are applets, wherein said conversion step further includes:

a preconversion substep for converting Java card IDs contained in said Java card CAP file into symbolic names, and for converting said Java card CAP file into a standard Java format, to obtain a preconverted file; and

a mapping substep for replacing in said preconverted file externally defined names with original names by using a mapping scheme between Java names and tokenized identifiers, to obtain the converted Java code file for a language-verification step; and

b) a language-verification step for verifying said converted Java code file for compliance with Java language specifications.

2. (cancelled)

3. (previously presented) The method for language verification of a Java card CAP file according to Claim 1, wherein said mapping substep is performed using a referenced Java export file which is available as a result of creating said Java card CAP file from said original Java code file.

4. (currently amended) The method for language verification of a Java card CAP file according to Claim 1, further comprising:

c) a signature step for creating, after verification of said converted Java code file in said language verification step, a [[second]] cryptographic signature file for the Java card CAP file.

5. (currently amended) The method for language verification of a Java card CAP file according to Claim 4, further comprising:

d) a loading step for loading the [[second]] cryptographic signature file to a [[storage device]] chipcard together with the Java card CAP file, wherein the cryptographic signature file is attached to the Java card CAP file when loaded in the chipcard.

6. (currently amended) The method for language verification of a Java card CAP file according to Claim 4, wherein the [[second]] cryptographic signature file is cryptographically verifiable, said method further comprising:

e) an executing step for executing said Java card CAP file upon a positive cryptographic verification.

7-9. (cancelled)

10. (previously presented) A computer-readable medium embodying computer program code, the computer program code comprising computer executable instructions configured for:

converting said Java card CAP file into a corresponding converted Java code file that is semantically identical to said Java card CAP file;

verifying said converted Java code file for compliance with Java language specifications;

converting Java card IDs contained in said Java card CAP file into symbolic names;

converting said Java card CAP file into a standard Java format, to obtain a preconverted file; and

replacing in said preconverted file externally defined names with original names by using a mapping scheme between Java names and tokenized identifiers, to obtain the converted Java code file.

11. (cancelled)

12. (previously presented) A computer-readable medium containing computer program code for a Java card CAP file language verifier for verifying a Java card CAP file that has been

derived from an original Java code file, said Java card CAP file including original Java semantics of said original Java card file, the computer program code comprising instructions for:

a converter for converting said Java card CAP file into a corresponding converted Java code file that is semantically identical to said Java card CAP file, wherein said converter further includes:

a preconverter for converting Java card IDs contained in said Java card CAP file into symbolic names, and for converting said Java card CAP file into a standard Java format, to obtain a preconverted file; and

a mapper for replacing in said preconverted file externally defined names with original names under use of a mapping scheme, to obtain the converted Java code file; and

a language verifier for verifying said converted Java code file upon its compliance with a Java language specification.

13. (cancelled)

14. (previously presented) The computer-readable medium according to Claim 12, wherein the mapper comprises an input for receiving a referenced Java export file created when a referenced Java card CAP file was converted from its corresponding original Java code file.

15. (previously presented) The computer-readable medium of Claim 12, wherein the instructions are further configured for a signature generator for generating a second cryptographic signature file.

16. (previously presented) The computer-readable medium of Claim 15, wherein the instructions are further configured for loading the second cryptographic signature file and the Java card CAP file to a storage device.

17. (previously presented) A computer-readable medium containing computer program code for a reduced file language verifier for verifying a reduced file that has been converted from an

original file, the reduced file maintaining original semantics of the original file, the computer program code comprising instructions for:

a converter for converting said reduced file into a corresponding converted file that is semantically identical to said reduced file, wherein said converter further includes:

a preconverter for converting IDs contained in said reduced file into symbolic names and for converting said reduced file into a standard format, to obtain a preconverted file; and

a mapper for replacing in said preconverted file externally defined names with original names under use of a mapping scheme, to obtain the converted file;

means for determining whether said reduced file complies with a predetermined language specification; and

a language verifier for verifying said converted file upon compliance with the predetermined language specification.

18. (cancelled)

19. (previously presented) The computer-readable medium of Claim 17, wherein said mapper comprises an input for a referenced difference file which is available as a result from a conversion in which a referenced reduced file has been converted from its original file.

20. (new) A method comprising:

converting an original file into a reduced file, wherein the original file contains a class description section and an instruction section, and wherein the reduced file contains a code description section that is based on the class description section, and wherein the reduced file contains a code section that is based on the instruction section, wherein the original file contains classes that are capable of being compiled, and wherein the only executable instructions in the reduced file are applets;

converting the reduced file into a converted file, wherein the reduced file and the converted file are semantically identical;

creating a cryptographic signature for the converted file; and

storing the cryptographic signature and the reduced file in a chipcard, wherein the cryptographic signature verifies that the reduced file was converted by a trusted entity.

21. (new) The method of claim 20, wherein the standard code file is a Java™ file, and wherein the CAP file is designed to be used by a Java™ card.

22. (new) The computer-readable medium of claim 10, wherein the Java card CAP file is created from an original file that contains classes that are capable of being compiled, and wherein the only executable instructions in the Java card CAP file are applets.